

[19]中华人民共和国国家知识产权局

[51]Int. Cl.<sup>6</sup>

G06F 15/00

G06F 17/30

## [12] 发明专利申请公开说明书

[21] 申请号 98800856.4

[43]公开日 1999年9月22日

[11]公开号 CN 1229489A

[22]申请日 98.6.16 [21]申请号 98800856.4

[30]优先权

[32]97.6.17 [33]US[31]60/049,853

[86]国际申请 PCT/US98/12557 98.6.16

[87]国际公布 WO98/58321 英 98.12.23

[85]进入国家阶段日期 99.2.14

[71]申请人 珀杜法尔玛 LP 公司

地址 美国康涅狄格州

[72]发明人 H·R·乌德爾 S·D·贝克

C·S·卡普爾 G·M·舍曼

W·里斯

[74]专利代理机构 上海专利商标事务所

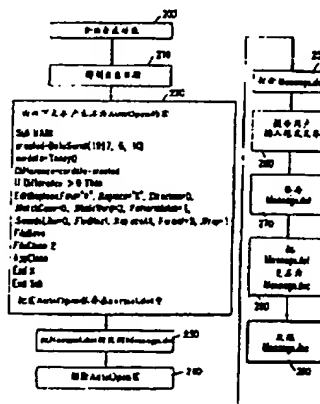
代理人 张政权

权利要求书 6 页 说明书 34 页 附图页数 13 页

[54]发明名称 自毁文档和电子邮件发报系统

[57]摘要

提供了一种自毁文档或电子邮件发报系统,该系统通过把宏或病毒附加到文档或电子邮件报文,以在预定的时间自动地销毁文档或电子邮件报文。产生(220)宏,并在它产生时把它附加(230)到诸如电子邮件报文(280)或文档(270)等文件。宏包含可执行代码或可执行程序的一部分,它指令计算机在所需的时间改写和/或删除附加了病毒的文件。



## 权 利 要 求 书

---

1. 一种创建自毁文档的方法，其特征在于所述方法包括以下步骤：  
创建一可执行模块，该模块指令计算机改写和/或删除所述可执行模块附加到其上的文档；  
将所述可执行模块附加到所述文档上。
2. 如权利要求 1 所述的方法，其特征在于：所述可执行模块是可执行的代码。
3. 如权利要求 1 所述的方法，其特征在于：所述可执行模块是可执行的程序。
4. 如权利要求 1 所述的方法，其特征在于：所述可执行模块是宏指令。
5. 如权利要求 1 所述的方法，其特征在于：所述方法进一步包括在打开所述文档时执行所述可执行模块的步骤。
6. 一种自毁电子邮件发报系统，其特征在于所述系统包括：  
可执行模块，该模块配置成指令计算机删除所述可执行模块附加到其上的报文；  
电子邮件发报系统，所述电子邮件发报系统配置成创建一电子邮件报文并发送所述电子邮件报文，所述电子邮件发报系统在发送前将所述可执行模块附加到所述电子邮件报文上。
7. 如权利要求 6 所述的系统，其特征在于：所述可执行模块是可执行的代码。
8. 如权利要求 6 所述的系统，其特征在于：所述可执行模块是可执行的程序。
9. 如权利要求 6 所述的系统，其特征在于：所述可执行模块是宏指令。
10. 如权利要求 6 所述的系统，其特征在于：所述可执行模块配置成采用空字符改写所述消息。
11. 如权利要求 6 所述的系统，其特征在于：所述可执行模块配置成指令所述计算机在出现预定条件时删除所述电子邮件报文。
12. 如权利要求 11 所述的系统，其特征在于：所述预定条件是指日期。

能的。

对计算机文件坚持文档保存政策的系统是众所周知的。通常，这些系统周期性地扫描网络中的文件并删除在特定日期前产生的文件。然而，由于这种系统在网络上进行操作，所以它们不可能删除存储在个人计算机的硬驱上的文件。为存取这些文件，在每个个人计算机上安装了文档保存程序。然而，即使在每个个人计算机上安装有文档保存系统，该系统仍旧不能删除存储在软盘或其它外部媒体上的文件。此外，这些系统不能删除已经由电子邮件转移到网络外部的计算机的文件。于是，例如，常规的文档保存系统不能删除已转移到雇员的家庭计算机或膝上型计算机或第三方的文件。

### 发明内容

依据本发明的第一实施例，提供了一种自毁文档和电子邮件发报系统，该系统通过把诸如“病毒”等可执行模块附加到文档或电子邮件报文以在预定时间自动地销毁文档或电子邮件报文。

计算机病毒是众所周知的。一般，计算机病毒是本身附加到宿主文件的可执行代码或程序的一部分。例如，“附加病毒”通过把自己附加到主程序的末尾并使主程序在执行主程序前执行病毒代码来进行操作。相反，“前置病毒”把自己附加到主程序的开始。其它类型的病毒位于主程序的内部。知道另一类病毒是“宏”病毒。这些病毒是嵌入文本文档中的宏，这些病毒可构成在每当打开、产生或保存文档时执行。通常，术语特洛伊马指属于其主文件或程序且不移动到其它文件或程序的病毒。

依据本发明的第一实施例，把具有特洛伊马形式可执行模块在文件(诸如电子邮件报文或文件)创建时附加到该文件。可执行模块包含可执行代码或可执行程序的一部分，它指令计算机在所需的时间改写和/或删除附加了可执行模块的文件。这样，可执行模块对文档或电子邮件报文的寿命加以限制。由于把可执行模块附加到文件，所以即使在把文件拷贝、转送或保存到盘片或磁带驱动器时，可执行模块也将随文件移动。

依据本发明第一实施例的另一个方面，每当打开附加了可执行模块的文件时就执行可执行模块。可执行模块确定是否已碰到文件删除的先决条件。如果碰到先决条件，则可执行模块指令计算机以零数据改写文件然后保存、关闭或删除该

文件。

这样，本发明提供了超过已有技术系统的许多优点。例如，由于把可执行模块附加到文件并在打开文件时执行，所以依据本发明的系统即使在已把文件存储在诸如软盘等外部媒体或非网络计算机上时也可坚持文档保存政策。这是因为可执行模块随文件移动且在无论何时何地每当打开文件时就执行。

〔依据本发明的第二实施例，提供了一种自毁电子邮件发报系统，该系统把可执行模块自动地附加到每个所产生的电子邮件报文或电子邮件报文附件。可执行模块包含可执行代码或可执行程序的一部分，它指令计算机在所需的时间改写和/或删除该可执行模块附加到的报文(和/或报文附件)。当把报文传输到其地址时，可执行模块随报文一起移动，这是因为可执行模块被附加到报文上。此外，即使在把报文拷贝、转送到另一个地址或保存到盘片或其它外部媒体时，可执行模块也保持附加于该报文。〕

依据本发明的第三实施例，一种文档安全系统包括其中“放置”有一个或多个数字对象(object)的虚拟容器。这样，术语“数字对象”被广泛地使用，它包括文档(诸如电子表格、图表/图形、字处理器文档、ASCII 文本文件、图像和其它数据文件)、程序以及可被存储在计算机上的任何东西。文档安全系统包括容器创建工具和容器开启工具。以在计算机上执行的一个或多个软件程序来实行容器创建工具和容器开启工具。容器开启工使用户可对容器中数字对象的有效寿命加以限制。例如，这些“寿命控制”可包括截止日期、截止日期和时间、文档可被打开的固定次数或其它限制。在由容器开启工具来打开该容器时，容器开启工具检查寿命控制。如果它们有效，则可访问包含在容器内的数字对象以使用户显示或执行。然而，如果寿命控制无效，则容器开启工具将立即销毁该数字对象。

包括容器创建工具和容器开启工具的文档安全系统的一个优点在于，在某些环境下，它与包含可执行模块的自毁文档相比，为寿命控制提供了更稳健的强制力。这是因为当前字处理和电子表格程序执行可执行模块的能力随产品的不同而变化很大。因此，例如，如果打开包括可由 Microsoft Word™ 执行的可执行模块的自毁文档则该文档将不能自毁，例如一个简单的 ASCII 文件没有构成识别和执行可执行模块。相反，由此文档安全系统，可把文档固定在由容器开启工具来打开的容器中，容器开启工具在使用时可对容器中的任何文档执行寿命控制，而文档的类型无关。

依据本发明第三实施例的文档安全系统的另一个方面，容器内的每个数字对象可具有独立的寿命控制。依据此特征，当容器内的单个物体期满时，可销毁该物体，而其它物体保持完整。

依据文档安全系统的再一个实施例，数字对象是依据本发明的第一或第二实施例产生的自毁文档。

依据本发明的又一个实施例，可抵挡希望破坏该安全系统的用户来保护容器、其内容及其寿命控制。通过使用加密技术来实行此保护。尤其是，使容器创建工具构成对容器内的物体加密，并使容器开启工具构成对物体解密。这样，如果用户打开一个物体而不利用容器开启工具，则该物体将不可读。

依据第一和第二实施例的另一个方面，通过可执行模块或通过其它工具对自毁文档或电子邮件报文进行加密，使可执行模块构成只在文档或电子邮件报文的寿命还未期满时对文档和电子邮件报文进行解密。

依据本发明的另一个实施例，通过把多个可执行模块嵌入文档或电子邮件报文来产生自毁文档，其中可通过不同的字处理或电子邮件系统来执行每个模块。例如，一个文档可包括可由第一系统执行的第一模块以及由第二系统执行的第二模块。文档本身可以是属于任一个系统的文档。依据此实施例，将执行对文档的寿命控制，而不管该文档是在第一系统还是在第二系统中打开。

依据本发明的再一个实施例，提供了一种利用虚拟容器的因特网商务系统。依据此实施例，希望在因特网上销售可以电子方式传输的产品的一方使用对产品加密并对产品设定时间控制的容器创建工具把产品放置在虚拟容器内。希望在购买产品前对产品看样的产品潜在购买者从销售者处获得容器以及容器开启工具的拷贝。容器开启工具使潜在购买者只能在预选的截止日期前才能观看和使用产品。在此实施例中，容器开启工具不能使用户从虚拟容器中除去该产品。依据此实施例的一个较佳方面，如果在截止日期后打开该容器，则由容器开启工具来删除该产品。依据此实施例的另一个方面，如果销售者在截止日期前接收到产品的付款，则销售者将把独有的密钥传输给购买者，可使容器开启工具构成在接收到此密钥时从容器中释放产品。】

#### 附图概述

图 1 示出可实行本发明的示意的已有技术环境。

图 2(a)和 2(b)示出依据本发明的电子邮件报文, 它们分别包括 pre-pending 和附加病毒。

图 3 是用于依据本发明的一个示意实施例产生自毁电子邮件报文的示意方法的流程图。

图 4 是把宏病毒嵌入图 3 的自毁电子邮件报文中的流程图。

图 5(a)到 5(c)示出用于依据本发明一个实施例的自毁文档的图形用户界面, 它是为 Microsoft Excel™ 文档而实行的。

图 6(a)到 6(c)示出用于依据本发明一个实施例的自毁文档的图形用户界面, 它是为 Microsoft Word™ 文档而实行的。

图 7(a)到 7(e)示出用于依据本发明一个实施例的自毁电子邮件报文的图形用户界面, 它是为 Microsoft Outlook™ 而实行的。

图 8(a)示出依据本发明一个实施例的文档组虚拟容器。

图 8(b)示出依据本发明一个实施例的虚拟容器代理。

图 9(a)和 9(b)示出本发明的文档组虚拟容器的另一个实施例。

图 10(a)示出构成保存单个文档的文档组容器的示意标题格式。

图 10(b)示出构成保存多个文档的文档组容器的示意标题格式。

图 11 示出用于容器创建工具和容器开启工具的图形用户接口。

图 12 是用于把文档加到图 10(b)的文档组容器的示意流程图。

图 13 是用于从图 10(b)的文档组容器中提取文档的示意流程图。

### 本发明的较佳实施方式

图 1 示出可利用本发明的示意的已有技术环境。局域网 1(LAN 1)包括多个办公计算机 10.1-10.6(以下集中叫做计算机 10)和服务 器 20。每个计算机 10 包括各个主存储设备 12(诸如硬驱)和各个辅存储设备 14(诸如软盘或 CD Rom 驱动器)。服务器 20 同样包括主网络存储设备 22(诸如硬驱)和辅网络存储设备 24(诸如磁带或 CD Rom 驱动器)。主和辅网络存储设备 22、24 上的数据是共享的, 它可被所有的计算机 10 所访问。相反, 每个计算机 10 的主和辅存储设备 12、14 上的数据是私有的, 它只能被其各自的计算机 10 所访问。服务器 20 经由传输线 50 对诸如家庭计算机 40 等网络外的计算机提供因特网访问。家庭计算机 40 包括主存储设备 42 和辅存储设备 44。LAN 1 支持电子邮件发报服务, 该服务使每个计算

机 10 在 LAN 1 内把报文发送到其它计算机 10，以及把报文向 LAN 1 外发送到诸如家庭计算机等网络外的计算机。图 1 所示的结构是商行中常用的典型 LAN 的示意。然而，本领域内的普通技术人员应理解，本发明可以各种网络结构来实行。此外，本发明的电子邮件发报系统还可等价地应用于非网络设备，例如，在这些设备中，由独立式计算机经由调制解调器或其它连接来发送和接收电子邮件报文。此外，可通过或不通过至外部计算机的调制解调器或其它连接而在独立式计算机上实行本发明的文档保存和删除特征。

现在将参考图 1 的结构来说明可依据本发明来减轻的文档保存和销毁物体的一些问题。

为了说明，假设把局域网安装在采用要求销毁超过两年的旧文档的文档保存政策的商行中。通常，通过在服务器 20 上安装周期性地扫描主网络存储设备 22 的程序并删除创建日期比当前扫描日期早两年多的文档来实行这种文档保存政策。这种程序的一个问题是，它没有充分地考虑到未存储在网络存储设备中的文档拷贝。

例如，假设用户-1 在计算机 10.1 上创建一文档，并把它存储在网络存储设备 22 上。然后，在一周以后，用户-1 把文档拷贝到计算机 10.1 的硬驱 12.1 上以及经由辅存储设备 14.1 拷贝到软盘上。此外，由于该文档在主网络存储设备 22 上，所以用户-2 可把该文档拷贝计算机 10.2 的主存储设备 12.2 上以及经由辅存储设备 14.2 拷贝到另一个软盘上，而不必让用户-1 知道。这样，已产生了该文档的五个拷贝，网络文档保存程序只能访问和删除其中之一。

由图 1 的 LAN 提供电子发报业务这一事实使上述问题更为复杂。结果，用户-1 可创建一电子邮件报文并经由因特网服务器 30 把它向 LAN 外发送到计算机 40。此外，可把先前创建的文档文件作为附件插入电子邮件报文中。这样产生了另外的问题。例如，可把电子邮件报文及其附件向 LAN 外发送到不了解商行的文档保存政策的第三方。在任何情况下，即使只把电子邮件发送到雇员的家庭计算机，该家庭计算机也不能被网络文档保存程序所访问。

通常，对未存储在网络存储设备上的文档执行文档保存政策需要每个雇员搜索其各自硬驱的每个目录以及其每个软盘或其它辅助存储媒体。此外，由于电子邮件系统可包括文档作为电子邮件报文的附件的能力，所以可把较旧的文档隐藏在最新的电子邮件报文中，使得难于识别和删除这些文档。雇员还必须搜索其膝

上型和家庭计算机中的商业文件。此外，即使每个雇员都能勤快地进行这些费时的任务，也将存在文档继续留在第三方的计算机上的可能性。

与实行文档保存政策有关的另一个问题是，通常，在从存储媒体删除文档时，并未除去该文档本身。相反，计算机只除去表示该文档在存储媒体中的位置的文档指针。文档本身仍保留在存储媒体中，直到计算机以另一个文档对其进行改写。结果，为了使雇员确保文档确实被删除，他们必须以空字符来改写该文档然后删除文档，从而进一步增加了与执行文档保存政策有关的时间。

### 嵌入文档或电子邮件报文中的可执行模块

依据本发明，提供了一种把可执行模块附加到文档的自毁文档系统。依据本  
10 发明的另一个实施例，提供了一种在电子报文或报文附件产生时把可执行模块附加到报文或报文附件的自毁电子邮件发报系统。可以任何公知的方式把可执行模块附加到报文或文档。例如，参考图 2，可把可执行模块作为前置病毒(图 2A)、作为附加病毒(图 2B)或以任何公知的方式来附加。例如，可利用汇编语言、高级  
15 编程语言或宏等任何方式来产生模块本身。附加的可执行模块使文档或报文在碰到预定条件时就自毁。这样，可执行模块设定了可控制文档寿命的“寿命控制”。例如，在上述说明中，对可执行模块进行编程，以在当前日期比文档或报文的创建日期晚两年多时删除文档或该文档附加到的报文。依据本发明的再一个实施例，可执行模块在删除文档或报文前以空字符(例如，全都是“X”)来改写文档或该文档附加到的报文。

20 依据本发明的还有一个实施例，可在出现其它或附加条件时判断改写和/或删除文档或报文。例如，系统可构成在文档 i)被关闭或 ii)被第二次打开时改写和/或删除报文或文档，从而产生在被销毁前只可读一次的文档。此外，系统还可构成在尝试拷贝、转送或打印报文或文档时改写和/或删除报文或文档。此外，系统还可构成阻止打印或拷贝报文或文档的任何尝试，从而产生只能看而不能打印的  
25 文档。还可以一事件来触发这些附加的寿命控制，从而在输入密码或密钥时解除阻止打印或拷贝，或者从而在预定的时间周期或其它事件后施加阻止打印或拷贝。

依据本发明的另一个实施例，该系统可使 LAN 管理者预定改写/删除条件，从而给所有的系统用户以及由系统用户所创建的所有文档或报文施加文档保存政策。同样，可使该系统构成使每个用户选择特定文档或报文是否包括改写/删除条件。此外，可使用户从各自改写/删除条件中进行选择。例如，这可以通过工具  
30



菜单来实现。

现在将描述用于实行自毁电子文档或提供自毁电子报文系统的示例方法。虽然这里所描述的示例实施例是针对 Microsoft Word™、Microsoft Excel™ 和 Microsoft Outlook™ 环境来加以说明的，但本领域内的普通技术人员应理解可在各自环境下以及以各自方式来实行本发明。

图 3 和 4 示出用于 Microsoft Word™ 6.0 文档的自毁电子发报系统的示意图。参考图 3，在步骤 200，用户开始在例如图 1 的办公计算机 10.1 上创建一文档或报文。在步骤 220，系统创建“AutoOpen”宏并把该宏保存在“Normal.dot”文件中。依据 Microsoft Word™ 的构造，每当打开 Word 文档时，就将执行标题为 AutoOpen 的宏。把步骤 220 处所参考的以 WordBasic™ 编程语言来写的指令作为 AutoOpen 宏保存在 normal.dot 模板文件中。Normal.dot 是由 Word™ 程序指定为全局宏的存储文件的模板。在步骤 230，把包括 AutoOpen 宏的 normal.dot 文件拷贝到名为“message.dot”的文件。然后，在步骤 240，从 normal.dot 文件中删除 AutoOpen 宏。在步骤 250-260，打开 message.dot 文件，并提示用户插入文档或报文的文本。然后，在步骤 270，保存现在包括文档或报文以及 normal.dot 文件的拷贝。message.dot 文件，normal.dot 文件的拷贝继而包括 AutoOpen 宏。在步骤 280-290，把 message.dot 文件更名为 message.doc，然后把它作为电子邮件报文或电子邮件报文附件发送。

〔在因特网上经由 LAN 服务器 20 和因特网服务器 30 把电子邮件报文发送家庭计算机 40。在由报文的收件人打开 message.doc 文件时，将执行嵌入 message.doc 文件的 AutoOpen 宏。图 4 是被嵌入 message.doc 文件的 AutoOpen 的流程图。在步骤 310，DateSerial()函数返还一表示报文创建日期的系列值作变量“created”。在创建 AutoOpen 宏的同时(图 3 的步骤 210 和 220)就设定函数 DateSerial 的自变量，把该自变量任意地表示为图 3 和 4 中的 1997 年 6 月 10 日。在步骤 320，Today()函数返还表示当前日期的系列值作为变量“curdate”。由该函数所返还的系列值为 1 和 802074 之间的整数，1 相应于 1899 年 12 月 31 日，802074 相应于 4095 年 12 月 31 日。在步骤 330，把变量“difference”设定为“curdate”与“created”之差。结果，变量“difference”等于从产生 message.doc 文件到打开文件的当前日期之间所经过的天数。

步骤 340 到 360 构成了一“If,Then”语句，它确定创建日期和当前日期之间

当打开包含嵌入程序的 Excel™ 文档时，执行 auto\_open 宏(表 1 的 2-20 行)，从自定义文档特性中读取“lcmode”和“ladata”的值(表 1 的 4-7 行)。如果“lcmode”等于“1”(第 9 行)且当前日期在“ladata”中的日期以后(第 10 行)，则把该文档的名称存储在变量“fn”(第 12 行)中，把声明该文档不再有效的报文作为报文框显示在计算机的显示屏幕上(图 5(c)和表 1 的 14 行)，并关闭和删除该文档(第 15-16 行)。虽然表 1 的程序在 Excelwd 被删除前不以空数据来改写该文档，但例如使用如下所述 Visual Basic™ 程序中的 Clear 方法可容易地加上此功能：

```
If DateValue(Date)>=DateValue(ladata) Then
Range("A1").Select
Range(Selection,Selection.SpecialCells(XlLastCell)).Select
Selection.Clear
ActiveWorkbook.Save
```

图 6a 到 6c 和表 2(a,b)示出一自毁文档系统，该系统是相对于 Microsoft Word 97™ 示出的。参考表 2(a,b)，示出用于产生自毁 Word 97 文档的示意的 Visual Basic™ 程序，该程序包括为 Microsoft Windows 环境而实行的用户界面。所示的示例利用 FileSave 和 FileSaveAs 宏来启动寿命控制，并利用 AutoOpen 宏来执行寿命控制。依据所示的实施例，把表 2 的程序作为模板嵌入文档中。

每当产生 Word 97™ 时，就把表 2 的程序嵌入该文档中。该程序位于作者的工作站上的外部文档模板中，并使用表 2c 所示的 AutoNew 宏拷贝到新的文档中。这个宏执行函数“InstallSDD”，该函数从模板 SDD.dot 中读取表 2 的程序并使用 Microsoft Word Organizer 对象把该程序的内容拷贝到新的文档中。依据本发明的实施例，使用此方法保证了把嵌入的程序封装到文档中。

当保存新文档(表 2b 的 FileSave 或 FileSaveAS)时，读取变量“lcmode”。如果未定义“lcmode”(即，这是第一次保存该文档)，则调用例程“pfImplementLifetimeControls”(表 2a)，且该程序询问用户是否希望实行寿命控制(图 6a，表 2)。如果他们希望，则该程序请求用户输入文档的期满日期(图 6(b)和表 2a)。然后把期满日期存储为“ladata”并把“lcmode”设定为“1”(表 2a)。如果用户不希望实行寿命控制，则把“lcmode”设定为“0”并把“ladata”设定为“”(表 2a)。把“lcmode”和“ladata”的值作为文档变量存储在 Word 97™ 文档中。

当打开包含嵌入程序的 Word 97™wd 时，执行 AutoOpen(表 2b)。使用

“pfGetLcmode”和“pfGetLcdata”函数(表 2a,2b)来读取“lcmode”和“lcdata”的值。如果“lcmode”等于“1”(表 2b), 且当前日期在“lcdata”中的日期以后, 则把该文档的名称存储在变量中(第 12 行), 把变量“rng”设定为文档的长度, 以空数据来改写文档(ret=rng.Delete)然后删除文档(Kill(dlg.name)), 并把声明该文档不再有效的报文作为报文框显示在计算机的显示屏幕上(图 6(c)和表 2b)。

图 7(a)到 7(e)以及表 3 示出一自毁文档系统, 该系统是相对于 Microsoft Outlook™ 示出的。参考表 3, 示出用于产生自毁 Outlook™ 电子邮件报的示意的 Visual Basic 程序, 该程序包括为 Microsoft Windows 环境而实行的用户界面。参考图 7(a)和 7(b), 把名为“Self-destruct”73 的制表符加到默认的电子邮件报文模板。图 7(a)示出在用户希望创建电子邮件报文时显示的常规“message”制表框 74。制表框 74 包括电子邮件报文文本 76 和电子邮件接收者的地址 77。图 7(b)示出自毁制表框, 它包括表示报文是否为自毁报文的触发器框 71 以及其中输入销毁日期的日期框 72。如下所述, 把销毁日期嵌入电子邮件报文中并在打开电子邮件报文时检索和处理。

Outlook 98 支持 Visual Basic 代码的 Outlook Items(诸如报文、日历入口、任务等独立对象), 为每个 Item 定义事件集合, 并可把 Visual basic 代码写为与每个事件一致。在 Outlook 对象模型中, 公知的电子邮件报文为“MailItem”。Outlook™ 使编程人员可截取和修改应用行为, 并截取和修改报文创建和报文打开行为。

表 3 示出 Visual Basic 程序, 该程序被嵌入电子邮件报文中并在使用 Item\_Open 函数打开电子邮件报文时执行。当产生电子邮件报文时, 把一检查输入自毁制表符 73 的触发器框 71, 并把一日期输入日期框 72, 然后把触发器框 71 的值存储在“Item.UserProperties(“DoDestruct”)”中, 并把输入日期框 72 中的日期存储“Item.UserProperties(“DestructDate”)”中。把这些值与表 3 的程序一起嵌入电子邮件报文中。当打开此电子邮件报文时, 自动地执行嵌入的程序 Function Item\_Open。参考表 3, 如果 DoDestruct 特性为 False(即, 该报文不是自毁报文), 则退出该程序并打开电子邮件报文。然而, 如果 DoDestruct 特性为 True, 则程序通过实行函数“itemExpired”来查看该报文是否期满, 继而实行函数“daysTilDestruct”。此函数 days TilDestruct 从“DestructDate”的值中减去当前日期并返还日期的差值。把由函数 days TilDestruct 返还的值存储在函数 itemExpired 的变量“dt”中。如果“dt”小于或等于零, 则函数“itemExpired”向函数 Item\_Open 返

现在将描述利用虚拟容器的依据本发明的文档安全系统。

依据图 8a 所示文档安全系统的第一实施例，文档组容器 500 用于存储一个或多个文档 510(或其它数字对象)，每个文档具有独立的寿命控制 520，在用户希望打开和显示容器内容的任何计算机上安装一工具。该工具使文档的寿命控制在把文档显示给用户前有效。如果寿命控制有效，则从容器中提取的文档可被用户所使用。如果寿命控制无效，则例如通过以空数据改写文档来销毁文档。

依据图 8(b)所示文档安全系统的第二实施例，容器代理 500'可用于存储单个文档 510'(或其它数字对象)。把代理处理程序安装在用户希望打开并显示代理容器中的文档的任何计算机上。与文档组容器不同的是，代理的操作对于用户是不可见的——除非文档的寿命控制 520'无效。为了提供此透明度，对代理进行命名或编码而显示为文档的普通实例，可改变把文档与其应用程序相关联的操作系统机构以使文档改为与代理处理程序相关联。如果寿命控制有效，则代理处理程序将过渡性地完成文档与应用程序之间的关联。本实施例利用了这样的优势，即在例如通过鼠标器单击“My Computer”中 Windows 95 文档图标、Windows 95™ 桌面中的文件或 Windows 3.1 的文件管理器中的文件来访问文件时，操作系统将尝试把该文件与诸如 MicroSoft™ Word、Microsoft Excel™ 或 WordPerfect™ 等应用程序关联。依据容器代理，把容器代理工具插在操作系统和应用程序之间，从而在用户访问容器代理(它将作为普通文档呈现给用户)时，操作系统将调用容器开启工具来自动地打开该容器，如果寿命控制有效，则将启动该应用程序并打开该容器中的文档。如果寿命控制无效，则例如通过以空数据改写文档和/或删除文档来销毁该文档。依据容器代理的另一个实施例，系统可构成在应用程序被关闭时把文档自动地返还容器代理。

现在将讨论文档组容器(DS 容器)的各种实施例。依据 DS 容器的第一实施例，提供了容器开启工具和容器创建工具。容器创建工具位于用户希望创建 DS 容器的计算机上，此工具的名称暗示着它可用于创建 DS 容器、指定寿命控制并把文档置于 DS 容器中。容器开启工具位于用户希望打开 DS 容器的计算机上，此工具可用于打开 DS 容器、检查寿命控制的有效性以及在寿命控制有效时从 DS 容器中提取文档。如果寿命控制无效，则例如通过以空数据改写文档来销毁该文档。依据一个实施例，DS 容器只包含一个文档。依据另一个实施例，可在单个容器中存储多个文档，每个文档可具有它自己的寿命控制。在想要使用户可打开

DS 容器而不是创建 DS 容器时,可把容器开启工具而不是容器创建工具安装到计算机上。未打开的 DS 容器作为单个文件呈现给用户,它包括标题信息(例如,包括寿命控制)以及该文档或这些文档本身,而与 DS 容器中所存在的文档数目无关。

依据本发明的多文档 DS 容器的另一个方面, DS 容器可包括容器标题,此标题包含该容器的控制信息并引用两个流的单链接列表。第一链接列表包含活动(active)流,第二链接列表包含非活动流。活动流相应于仍具有有效寿命控制的文档,非活动流相应于具有无效寿命控制或已从容器中除去(删除)的文档。活动和非活动流都具有标题,这些标题包含用于其相应流的寿命控制(以及最好是如下所述的加密信息)。这些流的标题中的每一个都包含对其列表(无论是活动的还是非活动的)中下一个流的引用指针(reference)或表示该流是其列表中最后一个流的指示符。每个流是二进制数据的连续块。

在使一个流无效或从容器中除去时,把其流标题标记为非活动并移动到非活动流列表中,并以空数据来改写该流。在把新的流加到容器中时,可在非活动流的长度足以包含此新的流时重新激活并使用此非活动流。如果没有非活动流或它不足以包含新的流,则可在容器中创建和激活新的流。如果容器的活动和非活动流列表经过许多变化,则该容器将变得碎片化。换句话说,在文档期满并被空数据替换时,活动文档可能被夹在只包括空数据的一个或多个非活动文档之间。在本发明的一个较佳实施例中,容器创建工具可检测此情况并对容器作必要的压缩。

图 9(a)和(b)示出本发明的活动/非活动流的实施例。图 9(a)示出已创建了三个文档包含在其中的虚拟容器。如图 9(a)所示,容器标题包括容器信息(诸如容器的名称、容器的默认寿命控制、容器创建的日期、最后修改容器的日期 DM 804 以及最后访问容器的日期)、与活动流有关的信息(诸如对活动流中第一个文档的指针和活动流中文档的数目)以及与非活动流有关的信息(诸如对非活动流中第一文档的指针以及非活动流中文档的数目)。容器中的每个文档包括文档标题,该标题包括寿命控制信息以及其各自流中下一个文档的指针。自从创建容器时,这三个文档就处于活动流中,容器标题包括活动流次标题,该次标题包括对第一个文档的指针以及表示这三个文档都处于活动流中的指示。非活动流次标题包括空信息。然后,在图 9(b)中,已把两个新的文档(文档 4 和 5)加到容器中,且文档 2

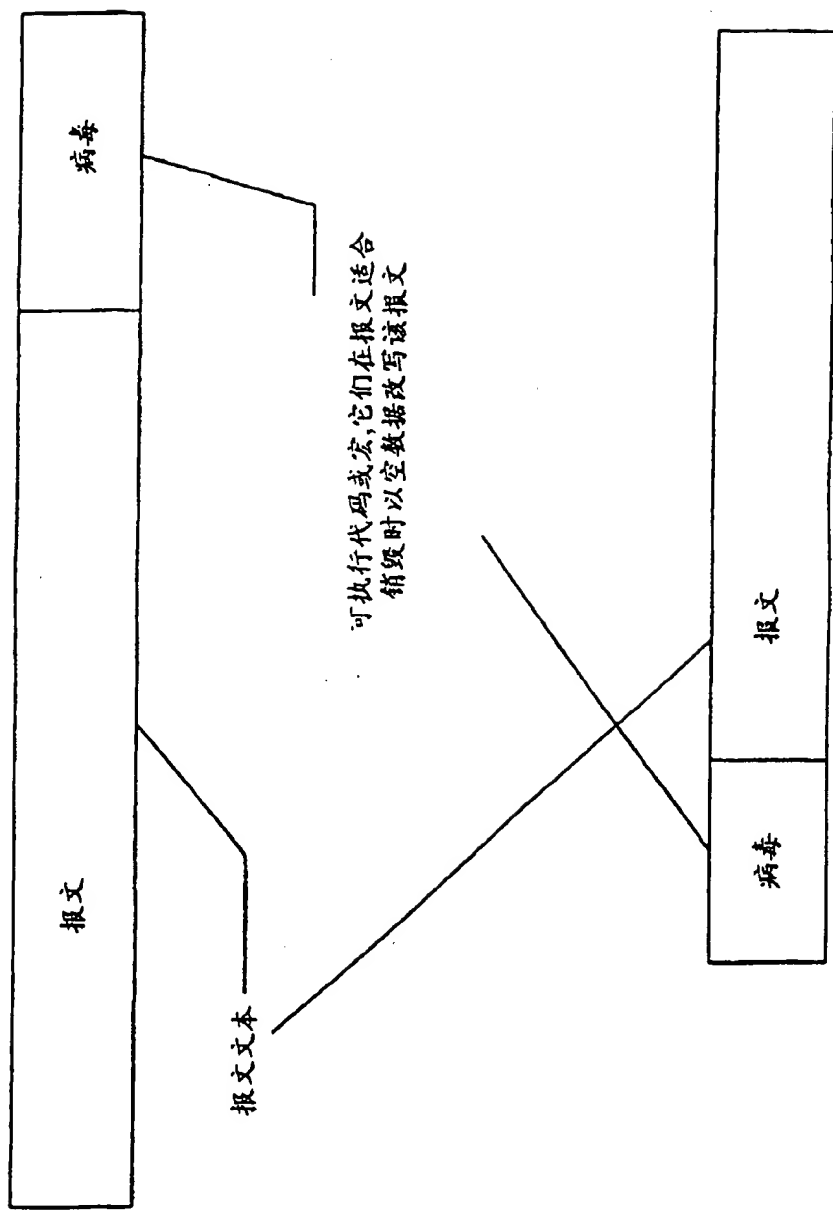


图 2